TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 11/17 pp70-76
Volume 12, Number 1, February 2007

Performance Analysis of Two Ethernet over E1 Schemes

CHEN Wentao (陈文涛), JIN Depeng (金德鹏), ZENG Lieguang (曾烈光)**

State Key Laboratory on Microwave and Digital Communication,
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

Abstract: The Ethernet over E1 approach, which takes advantage of widely deployed telecom networks, is an efficient and economical way to interconnect two Ethernets in different regions. Two Ethernet over E1 schemes, namely a byte granularity scheme and a frame granularity scheme are discussed. The byte granularity scheme partitions Ethernet frames into several pieces for transmission and has a strict requirement on the maximum delay difference of multiple E1 links. To solve this problem, the newly proposed frame granularity scheme transmits separately each frame through E1 links without any partitioning. The architecture designs of both schemes are presented. This paper evaluates the throughput and delay performances of both schemes, both analytically from results calculated from delay models and using test results from field programmable gate array (FPGA) implementation. Although the frame granularity scheme has a slightly worse delay performance, it has a higher throughput, and is the only choice able to overcome large delay differences of the E1 links.

Key words: Ethernet over E1; byte granularity; frame granularity; generic frame procedure (GFP); throughput; delay

Introduction

The Ethernet has undoubtedly become the most popular technique in local area network (LAN) for both its simplicity and low cost. The widespread usage of the Ethernet technology in LAN environments has forced the telecom operators to consider it as the only possible technology for metropolitan area network (MAN) access services to public and businesses[1]. Consequently, there has been much recent interest in Ethernet interconnection methods.

Several schemes have been proposed to take advantage of currently widely deployed synchronous optical network (SONET)/ synchronous digital hierarchy (SDH) networks. The Ethernet over SONET/SDH method was developed to connect two Ethernets in different regions using SDH[2]. The multi-protocol label switching (MPLS) protocol has been chosen by the Metro Ethernet Forum[3] to adapt Ethernet traffic to SONET/SDH networks, which is a sophisticated scheme which provides not only Ethernet interconnection but also Internet access. A scheme for connecting several Ethernets in a ring topology utilizing SONET/SDH links was subsequently modeled and analyzed using stochastic theory[4]. However, for users demanding less bandwidth, the Ethernet over E1 approach, which makes use of several E1 links to provide Ethernet interconnection, is more appealing, on account of both its low cost and ease of use. The device used in the Ethernet over E1 approach is called a reverse multiplexer, as it always adapts a high rate data stream to low speed channels.

The Ethernet over E1 scheme is shown in Fig. 1. Reverse multiplexer A connects LAN A to multiple leased E1 links. Reverse multiplexer B connects LAN

Received: 2005-11-25; revised: 2006-04-10
** To whom correspondence should be addressed.
E-mail: zenglg@mail.tsinghua.edu.cn; Tel: 86-10-62772387

B to the other end of the E1 links. The communication between two LANs is as follows. Reverse multiplexer A receives Ethernet frames from LAN A, packs them in several E1 frames which are then transmitted to E1 links. At the other end, Reverse multiplexer B receives E1 frames from E1 links, unpacks them to get the Ethernet frames transmitted from LAN A, and sends these frames to LAN B. Similarly, LAN B's Ethernet frames can be sent to LAN A. Thus, LAN A and LAN B are interconnected through multiple E1 links and two reverse multiplexers.

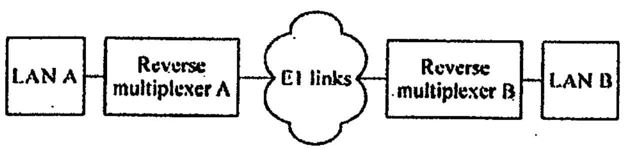

Fig. 1 Ethernet over E1

From the point of view of the reverse multiplexer, the directly connected LAN is called the local Ethernet, and the other LAN is called the remote Ethernet. Taking Fig. 1 as an example, LAN A is Reverse multiplexer A's local Ethernet, and LAN B is its remote Ethernet.

There are two Ethernet over E1 schemes, the byte granularity scheme[5] and the frame granularity scheme. The byte granularity scheme has already been widely used for several years, while the frame granularity scheme is newly proposed to fix the delay difference problem of the byte granularity scheme.

1 Two Ethernet over E1 Schemes

The byte granularity scheme and our newly proposed frame granularity scheme will be explained in this section. The essential difference between these two Ethernet over E1 schemes is in the granularity with which they manage multiple E1 links. We use Fig. 1 as the communication model in the following section. Here, assume that the number of E1 links leased is N.

In the byte granularity scheme, on receiving an Ethernet frame, Reverse multiplexer A partitions it to N pieces, and transmits them to different E1 links. At the other end, Reverse multiplexer B collects these N pieces, and recovers the Ethernet frame for LAN B. Due to delay differences between the E1 links, Reverse multiplexer B cannot recover the transmitted Ethernet frame until the last piece arrives. As a result, the reverse multiplexers have a strict requirement on the E1 links: the maximum delay differences of the E1 links cannot exceed the delay limitation, which is set by most designers as 8 ms or 16 ms[5].

Our newly proposed frame granularity scheme is designed to overcome the delay limitation of the byte granularity scheme. Instead of being partitioned into N pieces, each frame is wholly transmitted through one E1 link. On receiving an Ethernet frame from LAN A, Reverse multiplexer A selects one idle E1 link in a round-robin way, and sends the whole frame through that link. At the other end, Reverse multiplexer B collects each transmitted frame from each separate E1 link. The delay differences of the E1 links, therefore, have no impact on recovering the Ethernet frames, as the links independently transmit their own frames and do not need to wait for each other. A detailed scheduling algorithm will be given later, together with the required hardware architecture.

It must be mentioned that delay differences of the E1 links still affect the frame granularity scheme. Successive frames will suffer different delays as they travel on separate links, so that the order of the frames is not guaranteed in the frame granularity scheme. Fortunately, our experimental data reveals that Ethernet frame order is not the concern, as the receiver's media access controller (MAC) is able to reorder the frames for high layer applications. However, an impact on delay performance is inevitable, and it will be analyzed later.

2 Architecture Design

2.1 Hardware architecture

The reverse multiplexer is a device that connects a local Ethernet to multiple E1 links. On the Ethernet side, it interacts with the local Ethernet through a media independent interface (MII) as specified in IEEE standard 802.3[6]. On the E1 side, N E1 links are provided for communicating with a telecom network.

Figure 2 shows the hardware architecture of a reverse multiplexer, where the number of E1 links is set at 8. The scheme for mapping Ethernet frames to E1 frames operates as follows.

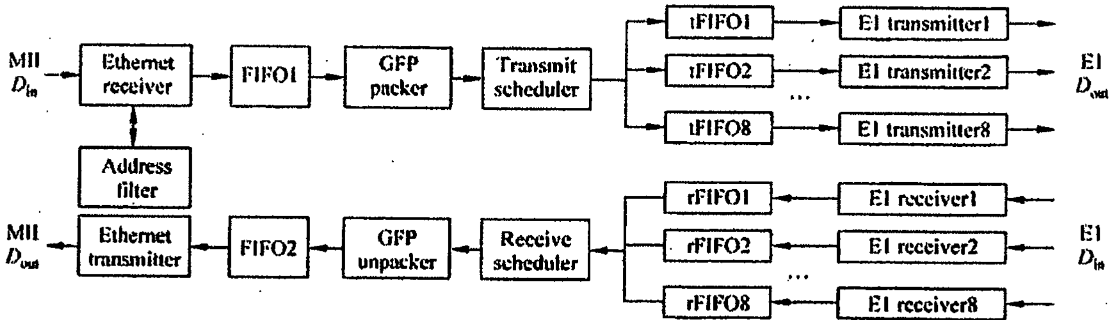

Fig. 2 Hardware architecture of reverse multiplexer

(1) The Ethernet receiver receives Ethernet frames through MII, verifies them for cyclic redundant check (CRC) errors, length errors, and dribbling bit errors, discards those with errors or those destined to the local Ethernet, and forwards the remaining frames for the remote Ethernet to FIFO1. The address filter manages the MAC address table with learning and aging functions. The table keeps all the MAC address entries of the local Ethernet by learning the source addresses of each frame. To save link bandwidth, Ethernet frames whose destination address is in the table are not transmitted to the E1 links, as they are destined for the local Ethernet. In addition, entries out of date are deleted from the table.

(2) The generic frame procedure (GFP) packer encapsulates the Ethernet frames according to the GFP specified in ITUT recommendation G.7041[7]. The encapsulated Ethernet frames, called GFP frames, are then sent to the transmit scheduler.

(3) The transmit scheduler assigns the GFP frames to eight E1 transmit buffers (tFIFO1-tFIFO8) in a round robin way. In the byte granularity scheme, once a byte is assigned, the transmit scheduler selects the next non-full buffer, and writes the next byte into it. In the frame granularity design, the transmit scheduler changes the link every frame. The transmit scheduler assigns the first frame to tFIFO1, the second frame to tFIFO2, and so on. In the byte granularity design, the transmit buffers are negligible, because each buffer only holds one byte. In contrast, in the frame granularity design, each buffer needs to be larger than the biggest GFP frame.

(4) The E1 transmitters (E1 transmitter1-E1 transmitter8) get the GFP data from their transmit buffers, respectively, pack them into E1 frames, and send to E1 links.

The reverse operation, mapping E1 frames to Ethernet frames, is carried out as follows.

(1) The E1 receivers (E1 receiver1-E1 receiver8) receive E1 frames from E1 links. In the byte granularity scheme, the E1 receivers store E1 frames to each receive buffers (rFIFO1-rFIFO8) for delay alignment. In the frame granularity scheme, the E1 receivers directly get the GFP frames from E1 frames, and store to their receive buffers, respectively.

(2) The receive scheduler reads data from the eight receive buffers, and sends the recovered GFP frames to the GFP unpacker, where the GFP frames are unpacked to get the Ethernet frames.

(3) The Ethernet frames forwarded to FIFO2 by the GFP unpacker are then read out by the Ethernet transmitter, and transmitted to the LAN through MII.

2.2 Design details

First, FIFO1 and FIFO2 are asynchronous FIFOs whose read and write clocks are not synchronous. Coding the memory addresses using gray codes helps to avoid full and empty misjudgments[8].

The second is the issue of virtual concatenation. In the byte granularity scheme, virtual concatenation technique in SDH[9] is used to synchronize the different E1 links. Thus, E1 frames are needed to carry the time information for the delay alignment, which increases the overhead.

The last thing to mention concerns the GFP encapsulation protocol. Several encapsulation protocols are currently in use, such as GFP, link access procedure-SDH (LAPS)[10], and point to point protocol (PPP)/high level data link control (HDLC)[11]. Experimental studies reveal that when the frame length is relatively short, the frame loss rate of the GFP and LAPS methods are nearly the same, in spite of the link bandwidth.

However, for longer frame lengths, the frame loss rate of the GFP method is much lower than that of LAPS[12]. Compared with PPP/HDLC, GFP does not inflate the data length in a non-deterministic manner, and has a more robust frame delineation mechanism[2]. All these traits are important in our choice of using GFP as our encapsulation protocol.

3 Performance Evaluation

3.1 Throughput performance

Each E1 frame has 32 slots, and each E1 link can carry a maximum data rate of 2.048 Mb/s. Thus, 8 links can provide a throughput of no more than 16.384 Mb/s.

In the byte granularity scheme, to accommodate the 8-ms delay difference, one extra slot is needed as a time stamp. The maximum throughput of the byte granularity system is, therefore, somewhat smaller than that of the frame granularity system. In our implementation, the 1st slot is used for E1 synchronization, the 2nd slot is used as time stamp, and the remaining 30 slots carry the payload. As a result, each E1 link can carry a data rate of 1.920 Mb/s, and the throughput of a device with 8 links is therefore 15.360 Mb/s.

In the frame granularity system, there is no need to carry time information in the E1 frames, so only the 1st slot is used for E1 frame synchronization. Thus, the data rate of each E1 link is 1.984 Mb/s, and 8 links provide a throughput of 15.872 Mb/s.

3.2 Delay performance

The ping program provided by the Windows® operation system is used to test the round trip delay of the system. Our analytical model is built on this basis, which serves as a guide to test the two schemes. The longest Ethernet frame is of 1518 bytes, and the maximum payload is of 1500 bytes, of which 8 bytes are IP mapping overhead from IP packet to Ethernet frames. Thus, an IP packet that is longer than 1492 bytes must be fragmented into several pieces in order to fit into the Ethernet container. Assume that the IP packet length is L in bytes, where $L \leqslant 65\,535$. If L is smaller than 1492, then this packet will be packed in one Ethernet frame. Otherwise, this packet will be fragmented into $\left\lfloor \frac{L}{1492} \right\rfloor + 1$ pieces, where $\lfloor x \rfloor$ means the largest integer smaller than x. Of these pieces, $\left\lfloor \frac{L}{1492} \right\rfloor$ pieces are of length 1492 bytes, and the remaining piece is of $L \pmod{1492}$ bytes. These Ethernet frames are then encapsulated in GFP frames for transmission to E1 links. This will introduce some GFP overhead. Here however, we neglect the GFP overhead for two reasons. First, GFP overhead is negligible for long IP packet lengths. Second, both the byte and frame granularity schemes require the same overhead. Neglecting the GFP overhead, therefore, has no impact on comparing these two schemes.

The number of E1 links is set at 8, as shown in Fig. 2. Figure 3 shows a delay model of the Ethernet over E1 system. Computer A sends an IP packet to Computer B. The packet is fragmented into several Ethernet frames, and transmitted to the Reverse multiplexer A. The transmit scheduler of Reverse multiplexer A assigns the traffic to eight E1 links either in byte granularity or in frame granularity, according to the scheme adopted. $t_{\rm ld1} - t_{\rm ld8}$ are the delays of the E1 links. For example, every bit transmitted through the 1st E1 link suffers a delay of $t_{\rm ld1}$ before it arrives at Reverse multiplexer B, where the Receive scheduler collects the frames transmitted by Computer A and forwards them to Computer B. Here, $t_{\rm D1}$ denotes total delay each bit suffers as it goes from Computer A to transmit scheduler of Reverse multiplexer A. Similarly, $t_{\rm D2}$ denotes the total delay from receive scheduler of reverse multiplexer B to Computer B.

According to the delay model above, each IP packet sent from Computer A to Computer B suffers three kinds of delay: the delay from Computer A to transmit scheduler of Reverse multiplexer A ($t_{\rm D1}$), the delay introduced by the E1 links, denoted as $t_{\rm ld}$, and the delay from receive scheduler of Reverse multiplexer B to Computer B ($t_{\rm D2}$). The total packet delay from Computer A to Computer B is

$$t_{\rm D} = t_{\rm D1} + t_{\rm ld} + t_{\rm D2} \tag{1}$$

Similarly, the packet delay from Computer B to Computer A is

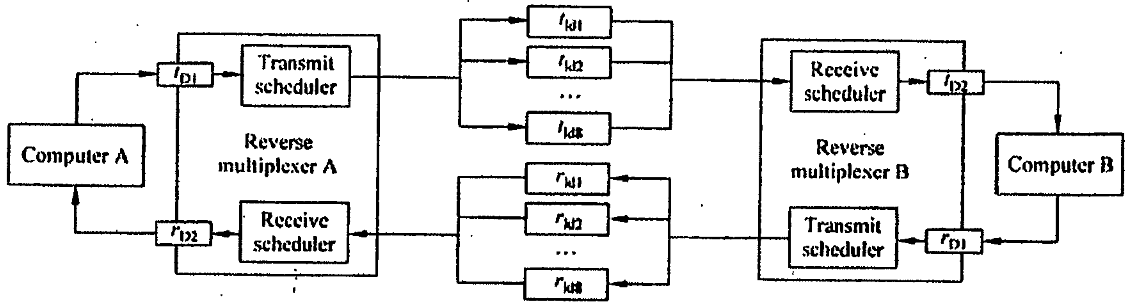

Fig. 3 Delay model of the system

$$r_D = r_{D1} + r_{ld} + r_{D2} \quad (2)$$

where r_{D1} is the delay from Computer B to transmit scheduler of Reverse multiplexer B, r_{ld} is the delay introduced by the E1 links, and r_{D2} is the delay from receive scheduler of Reverse multiplexer A to Computer A.

As a result, the round trip delay is

$$D = t_D + r_D = t_{D1} + t_{ld} + t_{D2} + r_{D1} + r_{ld} + r_{D2} \quad (3)$$

In the byte granularity scheme, the transmit scheduler of Reverse multiplexer A schedules the GFP frames in byte granularity. Assuming that the current byte goes to link i, where i is an integer between 1 and 8, and then the next byte goes to link j, where $j = i+1 \pmod 8$.

Thus, each 8 consecutive bytes go to different E1 links. Each GFP frame is thus approximately separated to 8 equal pieces, and each piece goes to the different E1 links. Now consider that an IP packet of length L is transmitted by Computer A. After fragmentation, it becomes $\left\lfloor \frac{L}{1492} \right\rfloor$ Ethernet frames with each of 1518 bytes, plus another Ethernet frame of $L \pmod{1492}+26$ bytes. These Ethernet frames are then encapsulated into GFP frames, segmented into 8 pieces, and transmitted to the E1 links. The receive scheduler of Reverse multiplexer B cannot recover the GFP frames until the last piece arrives. So the delay introduced by the E1 links is the maximum delay of all the links. According to Eqs. (1) - (3), we can obtain the packet round trip delay D_b in the byte granularity scheme as

$$t_{Db} = t_{D1} + t_{D2} + \max\ (t_{ld1}, t_{ld2}, \cdots, t_{ld8}) + \frac{L'}{R} \quad (4)$$

$$r_{Db} = r_{D1} + r_{D2} + \max\ (r_{ld1}, r_{ld2}, \cdots, r_{ld8}) + \frac{L'}{R} \quad (5)$$

$$D_b = t_{Db} + r_{Db} \quad (6)$$

where R is the total data rate of the E1 links, and L' is the number of bits of all GFP frames generated from the IP packet.

In the frame granularity system, the transmit scheduler of Reverse multiplexer A schedules the GFP frames in frame granularity. Each frame in a set of 8 consecutive frames goes to a different E1 link. Each GFP frame thus goes through a separate E1 link. Consider again that an IP packet of length L is transmitted by Computer A. After fragmentation, it becomes $\left\lfloor \frac{L}{1492} \right\rfloor$ Ethernet frames with each of 1518 bytes, plus another Ethernet frame of $L \pmod{1492}+26$ bytes. These Ethernet frames are then encapsulated into GFP frames, and assigned to 8 E1 links. The receive scheduler of Reverse multiplexer B collects these GFP frames from separate links, recovers the Ethernet frames, and sends them to Computer B. Each GFP frame goes through a separate E1 link, so some E1 links may be idle, when there are less than eight frames to be transmitted. In this scheme, the E1 links independently transmit their own GFP frames. Frames going through different links suffer different delays.

First, calculate the amount of data assigned to each E1 link. Based on the link delays, we can determine which link is the latest to forward all its traffic to Computer B.

Let $j = \left\lfloor \frac{L}{1492} \right\rfloor + 1$, where j is the number of GFP frames an IP packet of length L will be encapsulated into. In the scheme, these frames are assigned to eight links in a round robin way. Define L_i as the number of bytes assigned to link i. Then L_i is given by the following equations:

$$L_i = \begin{cases} 1518(c+1), & 1 \leqslant i < i_0; \\ 1518c + L(\text{mod } 1492) + 26, & i = i_0; \\ 1518c, & i_0 < i \leqslant 8 \end{cases} \quad (7)$$

with c and i_0 given by

$$c = \left\lfloor \frac{j-1}{8} \right\rfloor,\ i_0 = (j-1)(\text{mod } 8) + 1.$$

The packet delay is the maximum delay of all links. According to Eqs. (1)-(3), we get the packet round trip delay D_f in the byte granularity scheme:

$$t_{\mathrm{Df}} = t_{\mathrm{D1}} + t_{\mathrm{D2}} + \max\ (t_{\mathrm{ld1}} + \frac{L_1}{r}, t_{\mathrm{ld2}} + \frac{L_2}{r}, \cdots, t_{\mathrm{ld8}} + \frac{L_8}{r}) \quad (8)$$

$$r_{\mathrm{Df}} = r_{\mathrm{D1}} + r_{\mathrm{D2}} + \max\ (r_{\mathrm{ld1}} + \frac{L_1}{r}, r_{\mathrm{ld2}} + \frac{L_2}{r}, \cdots, r_{\mathrm{ld8}} + \frac{L_8}{r}) \quad (9)$$

$$D_f = t_{\mathrm{Df}} + r_{\mathrm{Df}} \quad (10)$$

where r is the data rate of each E1 link.

According to the data from the throughput analysis, let $R = 15.36$ Mb/s, $r = 1.984$ Mb/s, $t_{\mathrm{ld1}} - t_{\mathrm{ld8}} = 0$, $t_{\mathrm{D1}} = 0$, $t_{\mathrm{D2}} = 0$, $r_{\mathrm{D1}} = 0$, and $r_{\mathrm{D2}} = 0$. We can then determine the round trip delay of various IP packets, as shown in Fig. 4. For the same packet length, the byte granularity scheme has a slightly better performance than the frame granularity scheme, because the link bandwidth is better utilized in the byte granularity scheme, as links work together to transport the data, and no link will be idle if there are frames to be transmitted. However, in the frame granularity scheme, links transport the frames independently. Some links, therefore, might be idle, while others are working.

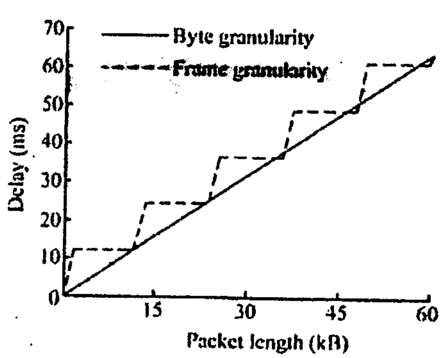

Fig. 4 Round trip delay

For certain specific packet lengths, there is less delay in the frame granularity scheme than in the byte granularity scheme, because the throughput of the frame granularity scheme is a little larger than that of the byte granularity scheme, where one extra slot is needed as a time stamp. Thus, IP packets with specific lengths can fully utilize the bandwidth of the frame granularity scheme, resulting in a smaller delay.

The last point to note concerning Fig. 4 is that for the same packet length, the packet delay in the frame granularity scheme does not exceed 12 ms, more than that of the byte granularity scheme, though it is larger in most cases.

4 Test Results

Field programmable gate array (FPGA) implementation of both schemes has been realized. The test environment is built as shown in Fig. 5. Two computers are connected using two reverse multiplexers plus 16 E1 cables. 16 E1 cables provide 8 E1 links.

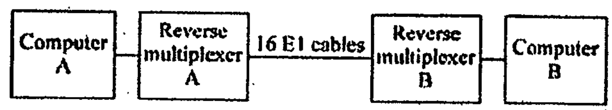

Fig. 5 Test environment

Computer A sends as many as possible Ethernet frames to Computer B, and we record the amount of data received by computer B as the throughput. The transmission of Ethernet frames of different lengths has been tested. The throughput results for both schemes are shown in Fig. 6. The results agree quite well with our expectations. The throughput is approximately 15.36 Mb/s for the byte granularity scheme, and 15.87 Mb/s for the frame granularity scheme.

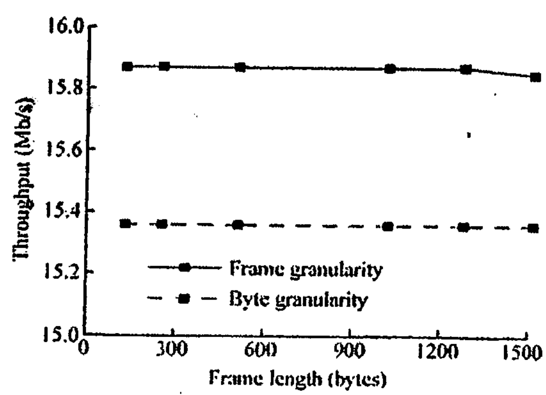

Fig. 6 Throughput results

The difference in delay time for the two schemes has also been investigated. For this, Computer A pings Computer B with packets of different lengths, and the round trip delays are recorded. The delay results for

the two schemes are shown in Fig. 7. The form of Fig. 7 is similar to that of Fig. 4, except for the cases of small packet lengths. The discrepancy is caused for two reasons. First, in calculating the data for Fig. 4, assume $t_{D1}=0$, $t_{D2}=0$, $r_{D1}=0$, and $r_{D2}=0$. These delays cannot, however, be neglected for small packet lengths. Second, the round trip delays are recorded by the ping program in units of milliseconds. The smaller the delay, the more significant will be the error in the data.

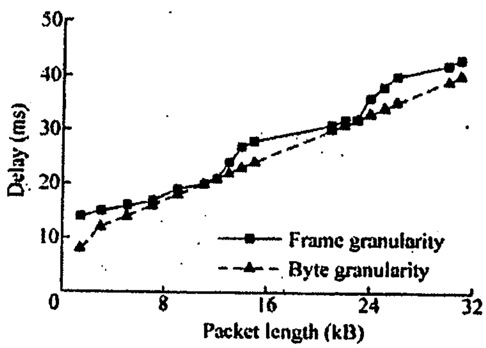

Fig. 7 Round trip delay results

5 Conclusions

Based on both the analysis and experimental results, we have compared two schemes for an Ethernet over E1 implementation. The conventional byte granularity scheme manages multiple E1 links at a finer granularity than the frame granularity scheme, and so it may be expected to perform better as a result. However, when the delay differences of multiple links are taken into account, the byte granularity scheme is somewhat punished for this finer management. First, the byte granularity scheme needs extra bandwidth to carry the time information, which results in a decreased throughput. To accommodate greater delay differences, more bandwidth and a buffer are required. Secondly, although the byte granularity scheme has a better delay performance, the delay difference between the two schemes does not exceed more than 12 ms for any given packet length.

As a result, each scheme has certain advantages, based on the link's quality and on the user's expectations of quality of service. When the delay differences of the multiple links are small, the byte granularity scheme offers better delay performance, whereas the frame granularity scheme offers a larger throughput. Otherwise, the frame granularity scheme is the only choice able to overcome large delay differences, while providing a bit higher throughput at the same time.

References

[1] Nenov G P C, Bigini G, Valente F. Transporting Ethernet services in metropolitan area networks (MANS). In: Proc. 12th IEEE International Conference on Networks. Singapore, 2004: 53-59.

[2] Li Xin, Jin Depeng, Zeng Lieguang. Encapsulation and rate adaptation for Ethernet over SDH. In: IEEE 2002 International Conference on Communications, Circuits and Systems and West-Sino Expositions. New York: IEEE Press, 2002: 1301-1305.

[3] Metro Ethernet Forum. http://www.metroethernetforum.org, 2004.

[4] Ge Liangwei, Jin Depeng, Zeng Lieguang. SDH link rate calculation for connecting multiple Ethernets. In: IEEE 2002 International Conference on Communications, Circuits and Systems and West-Sino Expositions. New York: IEEE Press, 2002: 1297-1300.

[5] Fan Yaxi, Ge Ning. Design and FPGA implementation of Ethernet over multiple E1 links. *Application of Electronic Techniques*, 2002, (11): 72-77. (in Chinese)

[6] IEEE 802.3. Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. New York, 2000.

[7] ITU-T Rec.G.7041. Generic framing procedure (GFP). Genevese, 2003.

[8] Sha Yanping, Huang Fuwei, Zeng Lieguang. VHDL design of asynchronous FIFO. *Application of Electronic Techniques*, 2001, (6): 74-75. (in Chinese)

[9] Shi Guowei, Wang Qing, Liu Zhao, Zeng Lieguang. SDH virtual concatenation technique used in Ethernet data transport. In: IEEE 2002 International Conference on Communications, Circuits and Systems and West-Sino Expositions. New York: IEEE Press, 2002: 1784-1787.

[10] ITU-TRec.X.X6/Y.1323. Ethernet over LAPS. Genevese, 2001.

[11] RFC 1662. PPP in HDLC-like framing. Sterling, VA, USA, 1994.

[12] Han Dahai, Li Li. Performance comparison of GFP and LAPS in application of data-transport. In: Proceedings of International Conference on Communications, Circuits and Systems. Hong Kong, 2005: 618-621.